GeneralVehicle

Release Documentation – June 10th, 2017
By Fred18

GeneralVehicle

Release Documentation – June 10th, 2017

By Fred18

Introduction

The General Vehicle Module is a simple module which will allow users to create and manage addons for Orbiter2016 which are meant to be vehicles, so not flying vessels.

Installation

Simply extract the content of the compressed file into Orbiter's main directory, letting it overwrite, if needed.

Compatibility

This addon is compiled against and compatible with Orbiter 2016 version.

Required Addons

No required addons

Keys

[NUMPAD8] = Accelerate

[NUMPAD2] = Brake

[NUMPAD4] = Turn Left

[NUMPAD6] = Turn Right

[NUMPAD5] = Set Steering Wheel immediately to center

[CTRL] + [NUMPAD4] = Slow Turn Left

[CTRL] + [NUMPAD6] = Slow Turn Right

[CTRL] + [NUMPAD2] = if vehicle is stopped it starts to move in reverse

[CTRL]+[L] = Toggle vehicle lights (local light sources option in Orbiter needs to be on)

[CTRL]+[G] = Grab the closest vessel available for towing (see relevant Towing section)

Note: Remember to be sure that the Num-Lock is activated in order for the numpad keys to work properly.

Release Policy

GeneralVehicle is distributed as FREEWARE. Its code is distributed along with the dll. Nobody is authorized to exploit the module or the code or parts of them commercially directly or indirectly.

You CAN distribute the dll together with your addon but in this case you MUST:

- Include credit to the author in your addon documentation;
- Add to the addon documentation the official link of Orbit Hangar Mods for download and suggest to download the latest and updated version of the module.

You CAN use parts of the code of GeneralVehicle, but in this case you MUST:

- Give credits in your copyright header and in your documentation for the part you used.
- Let your project be open source and its code available for at least visualization by other users.

You CAN NOT use the entire code for making and distributing the very same module claiming it as your work entirely or partly.

You CAN NOT claim that GeneralVehicle is an invention of yourself or a work made up by yourself, or anyhow let intend that it is not made and coded by the original author.

You install and use GeneralVehicle at your own risk, author will not be responsible for any claim or damage subsequent to its use or for the use of part of it or of part of its code.

Be also aware that I am not a professional programmer so code may be difficult to read and may present issues that I am not aware of.

Credits

Credits to Gattisplot, Donamy, Jacquesmomo, Jedidia, BrianJ, DaveS who kindly provided me with meshes and examples and gave me some of the ideas included in the module.

Meshes of F1 car and Ducati motorbike were taken from sketchup online available repository.

How to make a GeneralVehicle vessel

GeneralVehicle is based on a simple parameter settings inside the Vessel's configuration file, located in the folder x:\..\Orbiter_Root\Config\Vessels\.

In order to make a new vehicle, just browse to that folder, copy and paste an existing configuration file from another GeneralVehicle vessel, rename it and start to modify it as per your preferences. Once saved it will be available to place in any scenario, both editing manually the scenario or via Scenario Editor Dialog.

All you need for start is a mesh of your vehicle in which you know the mesh groups of the wheels and of the steering wheel (if present) . If you don't have the groups' numbers and you have D3D9 installed, you can easily obtain them using the D3D9 mesh debug dialog, in which you can "pick" groups and therefore take notes of their numbers directly in the sim.

Note: If wheels groups are not specified the module will work anyway. It can be used for making, for example, people walking, or other things moving on the ground without rotating wheels.

All the imputs of the cfg file

Item	Unit	Meaning	Default value
Empty_Mass = xxx	Kg	Empty Mass of the vehicle (basically unused)	1000
Mesh = abc	-	Mesh filename	Mandatory by User
Is_Motorcycle = TRUE/FALSE	-	If set to TRUE the vehicle will lean instead of turn the wheels	FALSE
Cockpit_Pos = xxx yyy zzz	m m m	Position of the inside cockpit cam	0 0 0
Acceleration = xxx	G	Acceleration value	1
Brake = xxx	G	Deceleration Value	1
Max_Speed = xxx	Km/h	Maximum Speed Value	a huge value…
Reverse_Max_Speed = xxx	Km/h	Maximum Speed in Reverse	half of Max_Speed
Always_Upright = TRUE/FALSE	-	If set to TRUE the vehicle won't follow the terrain and will always stay aligned with the planet's axis	FALSE
Steering_Speed = xxx	Deg/s	Value that expresses how quick the steering is applied	2
Max_Steering_Angle = xxx	Deg	Maximum steering value per side	45
Full_Pedal_Time = xxx	s	Time to have the full pedal of acceleration or brake fully pressed while pressing relevant key	0.01 (almost instant)
Rear_Right_Groups = xx yy zz tt …	-	Mesh groups of rear right wheel	-
Rear_Left_Groups = xx yy zz tt …	-	Mesh groups of rear left wheel	-
Front_Left_Groups = xx yy zz tt …	-	Mesh groups of front left wheel	-
Front_Right_Groups = xx yy zz tt …	-	Mesh groups of front right wheel	-
Rear_Axle_Pos = xxx yyy zzz	m m m	Position of rear axle	Calculated by the module from the mesh groups

Front_Axle_Pos = xxx yyy zzz	m m m	Position of front axle	Calculated by the module from the mesh groups
Steering_Axis = xxx yyy zzz	m m m	Rotation axis of the front steering	Calculated by the module from the mesh groups
Camber = xxx	Deg	Camber angle of the front wheels	0
Front_Wheels_Diameter = xxx	m	Diameter of front wheels	Calculated by the module from the mesh groups
Rear_Wheels_Diameter = xxx	m	Diameter of rear wheels	Calculated by the module from the mesh groups
Four_Wheels_Steering = TRUE / FALSE	-	If set to TRUE also the rear wheels will steer	FALSE
Rear_Steering_Axis = xxx yyy zzz	m m m	Rotation axis of the rear steering	Calculated by the module from the mesh groups
Middle_Axles_N = xxx	-	Number of middle axles	0
Middle_Groups_X = xxx yyy zzz ttt …	-	Mesh groups of the wheels of middle axle number X	-
Middle_Axle_Pos_X = xxx yyy zzz	m m m	Position of middle axle number X	Calculated by the module from the mesh groups
Middle_Wheels_Diameter = xxx	m	Diameter of middle wheels number X	Calculated by the module from the mesh groups
Steering_Wheel_Groups = xxx yyy zzz ttt …	-	Mesh groups of steering wheel	-
Steering_Wheel_Axis = xxx yyy zzz	m m m	Rotation axis of the steering wheel (if present)	Calculated by the module from the mesh groups
Height_From_Ground = xxx	m	Height from ground of the vehicle	Calculated by the module from the mesh
Listen_From = abc	-	Name of the vessel from which the vehicle listens to the key presses	-
Towing_Attachment_Pos = xxx yyy zzz	m m m	Attachment position for towing point	0 0 0
Tow_Max_Angle = xxx	Deg	Maximum angle between vehicle and towed vessel	160
Front_Lights_Pos = xxx yyy zzz	m m m	Front right running light position	0 1 3
Rear_Lights_Pos = xxx yyy zzz	m m m	Rear right running light position	0 1 -3
No_Steering_Animation = TRUE/FALSE	-	If set to TRUE the vehicle will turn but there will be no steering animations for the wheels	FALSE

Some notes on the parameters

Empty_Mass is a basically unused parameter, since GeneralVehicle forces the state update of the Vessel.

Acceleration and Brake values are in Gs, therefore they are planet dependent: on earth the 9.8 m/s2 is used, but for other planets the relevant gravity acceleration is used.

Always_Upright is used to avoid that the vehicle follows terrain inclination. If set to TRUE the vehicle will always be parallel to an hypothetical flat ground.

Whenever as default value it is said "Calculated by the module from the mesh groups" it means that the module will try to guess the values using the mesh coordinates. This of course can be inaccurate, so users have the chance to manually modify every parameter he wishes to. The default calculations are made in the following way:

Front/Rear/Middle_Axle_Pos → the module will calculate the barycenter of the front/rear/middle wheel groups and use that.

Steering_Axis/Rear_Steering_Axis → the module will calculate the barycenter of the front right /rear right wheel groups and use that (changing the sign of the x value for the left part).

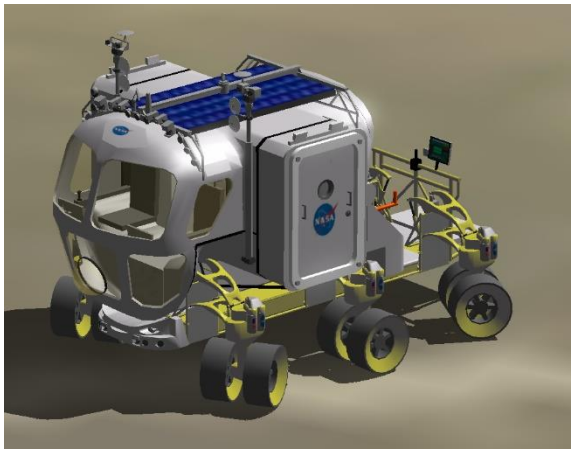
Front/Rear/Middle_Wheels_Diameter → the module will find the highest and the lowest points along y axis of the relevant mesh groups and will use the difference between them as wheel diameter.

Steering_Wheel_Axis → the module will use the barycenter of the steering wheel groups

Height_From_Ground → the module will look for the mesh lowest point along y axis, change it in sign and use that.

Max_Steering_Angle → if the vehicle is a motorcycle the front wheel won't steer and the value of Max_Steering_Angle will be used as maximum lean value for the motorbike.

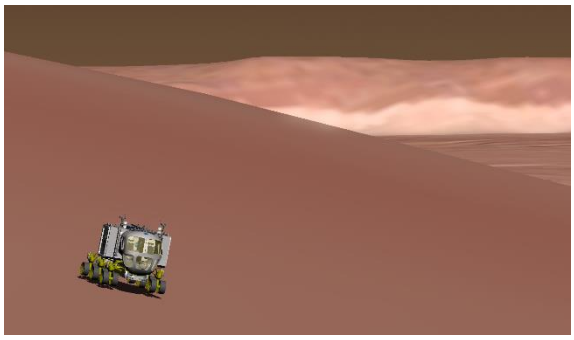
All the above values will be noted down in the orbiter.log file for user to see it! If you have troubles with coordinates, remember to check the log file, it may be of great assistance!



Four Wheels Steering Example

F1 Car Steering

Leaned Motorbike



LER on Mars, following terrain inclination

The Listen_From parameter

GeneralVehicle module gives the opportunity to have a vehicle "listening" from key presses coming from other vessels in the sim. This can be very useful if a user wants to have a vehicle with for example a series specific animations. Let's take the Mars Curiosity rover example: the user will be able to make a mesh with just the wheels of the rover, use that as GeneralVehicle, attach on top of it the

Curiosity rover module with for example robotic arms animations and everything he wants to, and then drive the rover directly from the upper part, without having to switch back and forth for movement.

Remember that the parameter must be the name of the vehicle in the sim, not the classname, not the module name, but the name in the sim.

To have other attachment points on the GeneralVehicle vessel, just add them in the cfg file with the regular syntax.

For example, with the LER examples provided you'll see in the GV_LERWHEELS.cfg file:

```
BEGIN_ATTACHMENT
C 0 0 0  0 1 0  0 0 1  BDY
END_ATTACHMENT
```

And in the LERBODY.cfg file:

```
BEGIN_ATTACHMENT
P 0.0 0 0  0 -1 0  0 0 1  BDY
END_ATTACHMENT
```

And in the relevant scenario:

```
LERWHEELS:GV_LERWHEELS
  STATUS Landed Moon
  POS -33.4747431 41.0690295
  HEADING 236.50
  ALT 2.166
  AROT -155.665 8.575 152.402
  AFCMODE 7
  NAVFREQ 0 0
END
LERBODY:LERBODY
  STATUS Landed Moon
  BASE Brighton Beach:1
  POS -33.4374968 41.1184013
  HEADING 66.99
  ALT 2.233
  AROT 17.341 -16.521 43.998
  ATTACHED 0:1,LERWHEELS
  AFCMODE 7
  IDS 0:560 100 1:564 100
  NAVFREQ 0 0
END
```

Important note: in the GeneralVehicle vessels attachments added through cfg file will count from 1, since number 0 is taken by the towing attachment point. This is why you see the line `ATTACHED 0:1,LERWHEELS`.

Towing

GeneralVehicle gives the opportunity to simulate the towing/pushing of other vessels in the sim.

In order to do this the vessels in the sim must have a proper attachment point defined. This can be done in their relevant cfg file. The syntax of the attachment point is specific. Here's an example for the DeltaGlider:

```
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 -2.37 8.535  0 0 1  0 1 0  TW_11.65
END_ATTACHMENT
```

In particular:

P = must be P because it is an attachment for a parent vessel

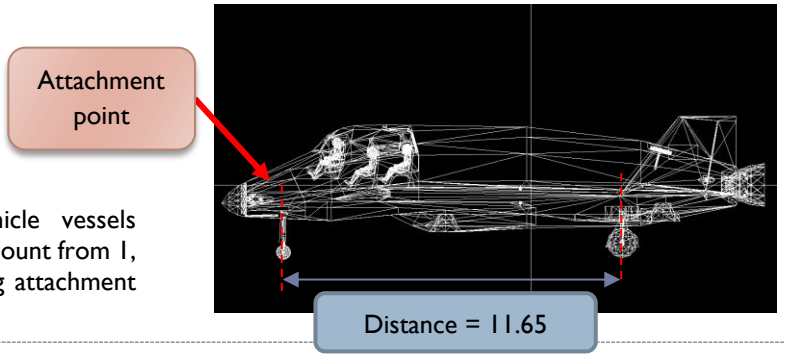
0 -2.37 8.535 = are the coordinates of the point

0 0 1 = direction of the attachment

0 1 0 = rotation of the attachment

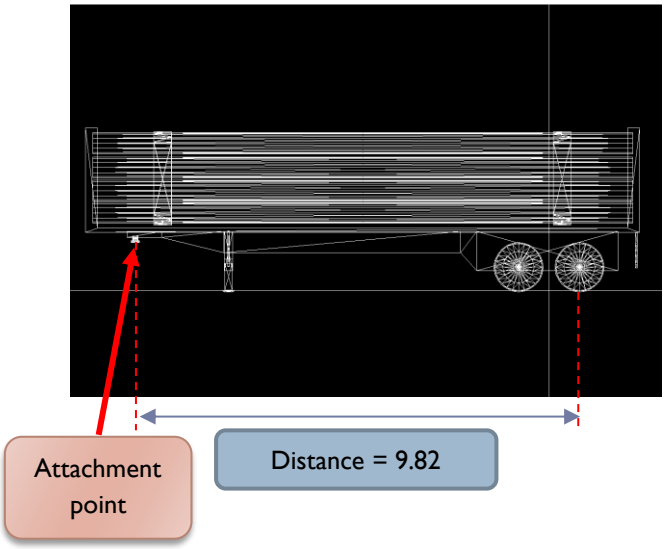
TW_11.65 = ID of the attachment.

In order to grab a vehicle for towing the attachment ID of the attachment point in the towed vessel must be in this format "TW_XX.XX". "TW_" is mandatory to identify the point. The numbers there indicate the distance between the attachment point and the rear axle of the towed vessel. That is necessary to have a correct towing effect! Some examples here below:



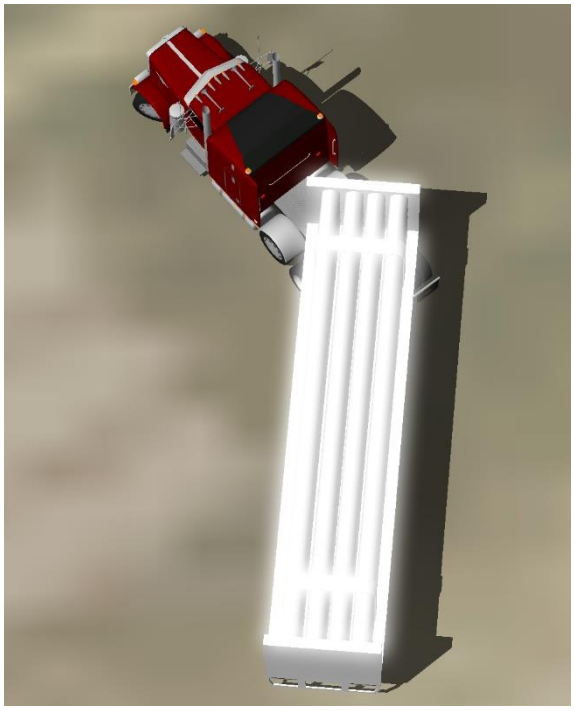
by Fred18

Relevant Attachment Line:
P 0 -2.37 8.535 0 0 1 0 1 0 TW_11.65



Relevant Attachment Line:
P 0 -0.05 9.123 0 0 1 0 1 0 TW_9.82

Once the attachment is properly defined, the vessel will be available for towing in the sim. The user must press [CTRL] + [G] to grab or release a vessel. The vehicle will look for all vessels available for towing in the sim, choose the closest one and check if it's within 50 meters from it. If it is, it will attach it automatically. To release just press [CTRL]+[G] again.



Thanks

Biggest thanks to Dr. Martin Schweiger for creating Orbiter!

Have fun up there, Fred18